

DETAILED ACTION

1. This communication is responsive to the amendment filed 12/19/2008 and the telephonic interview on 03/27/2009.

Claims 1-4, 6-13, and 17-22 have been examined and allowed.

2. **EXAMINER'S AMENDMENT:**

An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it **MUST** be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Ms. Annemarie Vicere (Registration No. 58, 965) on 03/27/2009.

The application has been amended as follows:

In the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method for loading a first kernel module, comprising:
 - loading a preload module into a kernel, wherein loading the preload module comprises creating a dynamic dependency list;
 - loading the first kernel module comprising a static dependency list into the kernel;
 - triggering a hook when the static dependency list is reviewed;
 - obtaining module information associated with the first kernel module using the hook;
 - searching, in the dynamic dependency list, for a dynamic dependency between the first kernel module and a second kernel module using the module information; [[and]]
 - updating the static dependency list with the dynamic dependency between the first kernel module and the second kernel module to obtain an updated static dependency list when the dynamic dependency between the first kernel module and the second kernel module is present in the dynamic dependency list,
 - creating an interposer module, wherein the interposer module defines a new kernel symbol definition for a kernel symbol;
 - defining a configuration file, wherein the configuration file indicates that the first kernel module is dependent on the interposer module;
 - loading the interposer module before the first kernel module using the static dependency list;
 - re-defining the kernel symbol using the new kernel symbol definition;

loading the first kernel module, wherein the first kernel module comprises
the kernel symbol; and
resolving the kernel using the new kernel symbol definition.

2. (Currently Amended) The method of claim 1, ~~further comprising:~~
~~defining a configuration file,~~ wherein the configuration file is read by the preload module to create the dynamic dependency list.
3. (Currently Amended) The method of claim 2, further comprising:
 - updating the configuration file to obtain an updated configuration file;
 - unloading the preload module from the kernel;
 - re-loading the preload module into the kernel, wherein re-loading the preload module comprises reading the updated configuration file to generate an updated dynamic dependency list.
4. (Original) The method of claim 1, further comprising:
 - installing the hook using the preload module.
5. (Cancelled)

6. (Currently Amended) The method of claim [[5]] 1, wherein the interposer module comprises a reference count.
7. (Original) The method of claim 1, wherein the hook is installed in a kernel runtime loader.
8. (Currently Amended) The method of claim 1, wherein the first kernel module comprises a flag indicating that [[a]] the kernel symbol may have multiple definitions.
9. (Original) The method of claim 1, wherein updating the static dependency list comprises adding the dynamic dependency to the static dependency list.
10. (Currently Amended) A system comprising:
 - a processor;
 - a first kernel module executing on the processor and having a static dependency list and a kernel symbol;
 - a modified kernel runtime loader configured to load the first kernel module using the static dependency list; and

a preload module comprising a dynamic dependency list, wherein the preload module is configured to install a hook into the modified kernel runtime loader,

an interposer module configured to provide a new kernel symbol definition for the kernel symbol, wherein the first kernel module is dependent upon the interposer module,

a configuration file configured to define a dynamic dependency, wherein the configuration file indicates that the first kernel module is dependent upon the interposer module,

wherein the hook is configured to update the static dependency list using the dynamic dependency list to obtain an updated static dependency list when [[a]] the dynamic dependency between the first kernel module and a second kernel module is present in the dynamic dependency list, and

wherein the kernel symbol is resolved using the new kernel symbol definition.

11. (Currently Amended) The system of claim 10, ~~further comprising:~~

~~a configuration file configured to define the dynamic dependency, wherein the configuration file is used to generate the dynamic dependency list.~~

12. (Original) The system of claim 11, wherein the configuration file is read by the preload module when the preload module is loaded.

13. (Original) The system of claim 11, wherein the preload module is unloaded and reloaded when the configuration file is modified.
- 14.-16. (Cancelled)
17. (Currently Amended) The system of claim 10 [[14]], wherein the interposer module comprises a reference count.
18. (Currently Amended) The system of claim 10 [[14]], wherein the first kernel module comprises a flag indicating that [[a]] the kernel symbol may have multiple definitions.
19. (Previously Presented) The system of claim 10, wherein the preload module searches the dynamic dependency list using first kernel module information to determine whether the dynamic dependency associated with the first kernel module is present.

20. (Original) The system of claim 10, wherein updating the static dependency list comprises adding the dynamic dependency to the static dependency list.
21. (Currently Amended) A computer system for loading a first kernel module, comprising:
- a processor;
 - a memory;
 - a storage device;
 - a computer display; and
- software instructions stored in the memory for enabling the computer system under control of the processor, to perform:
- loading a preload module into a kernel, wherein loading the preload module comprises creating a dynamic dependency list;
 - loading the first kernel module comprising a static dependency list into the kernel;
 - triggering a hook when the static dependency list is reviewed;
 - obtaining module information associated with the first kernel module using the hook;
 - searching, in the dynamic dependency list, for a dynamic dependency between the first kernel module and a second kernel module using the module information; [[and]]

updating the static dependency list with the dynamic dependency between the first kernel module and the second kernel module to obtain an updated static dependency list, when the dynamic dependency between the first kernel module and the second is present in the dynamic dependency list;

creating an interposer module, wherein the interposer module defines a new kernel symbol definition for a kernel symbol;

defining a configuration file, wherein the configuration file indicates that the first

kernel module is dependent on the interposer module;

loading the interposer module before the first kernel module using the static dependency list;

re-defining the kernel symbol using the new kernel symbol definition;

loading the first kernel module, wherein the first kernel module comprises the kernel symbol; and

resolving the kernel using the new kernel symbol definition.

22. (Currently Amended) A network system having a plurality of nodes, comprising:

a first kernel module having a static dependency list and a kernel symbol;
a modified kernel runtime loader configured to load the first kernel module using the static dependency list; and

a preload module comprising a dynamic dependency list, wherein the preload module is configured to install a hook into the modified kernel runtime loader,

an interposer module configured to provide a new kernel symbol definition for the kernel symbol, wherein the first kernel module is dependent upon the interposer module,

a configuration file configured to define a dynamic dependency, wherein the configuration file indicates that the first kernel module is dependent upon the interposer module,

wherein the hook is configured to update the static dependency list using the dynamic dependency list to obtain an updated static dependency list when ~~[[a]] the~~ dynamic dependency between ~~associated with~~ the first kernel module and a second kernel module is present in the dynamic dependency list,

wherein the kernel symbol is resolved using the new kernel symbol definition,

wherein the kernel module executes on any node of the plurality of nodes, wherein the modified kernel module runtime loader executes on any node of the plurality of nodes,

wherein the preload module executes on any node of the plurality of nodes, and

wherein ~~at least one of~~ each of the plurality of nodes comprises a processor.

CONTACT INFORMATION

Any inquiry concerning this communication or earlier communications from the examiner should be directed to VAN H. NGUYEN whose telephone number is (571) 272-3765. The examiner can normally be reached on Monday-Thursday from 8:30AM - 6:00PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, MENG-AI AN can be reached at (571) 272-3756.

The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair.direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/VAN H NGUYEN/

Primary Examiner, Art Unit 2194